

## **AMENDMENTS TO THE CLAIMS**

The following listing of claims will replace all prior versions and listings of claims in the application.

### **LISTING OF CLAIMS**

1. (currently amended) A method for controlling the value of a RAM variable inside an executable program, comprising:

presenting a software program in executable form and having a plurality of machine instructions of a finite quantity of fixed lengths;

identifying at least one machine instruction that accesses a variable defined in random access memory associated with the software program;

replacing the identified machine instruction in the executable form of the software program with a branch instruction that references an address outside an address space of the software program;

~~defining a replacement instruction for the at least one machine instruction; and~~

~~replacing the at least one machine instruction in the executable form of the software program with the replacement instruction;~~

defining a set of relocated instructions at the address referenced by the branch instruction, wherein the set of relocated instructions function to change a value of the variable; and

executing the executable form of the software program having the branch instruction.

2. cancel

3. cancel

4. cancel

5. (original) The method of Claim 1 wherein the step of identifying at least one machine instruction further comprises determining location information for the at least one machine instruction within the software program.

6. (original) The method of Claim 5 wherein the step of determining location information further comprises identifying an address for the at least one machine instruction using the image of the executable containing the machine instructions that comprise the executable.

7. (original) The method of Claim 6 wherein the step of replacing the at least one machine instruction further comprises inserting the replacement instruction into a program memory image of the software program at said address.

8. (currently amended) The method of Claim 2 wherein said branch instruction references a set of relocation instruction residing at ~~outside~~ an unused address space of ~~for~~ the software program.

9. cancel

10. (currently amended) A computer-implemented calibration system for modifying RAM variables of a software program embedded in a microprocessor, comprising:

an instruction locator operating on a different processor, the instruction locator adapted to receive an address for RAM variable within an software program and operable to identify at least one machine instruction in an executable form of the software program that accesses the RAM variable; and

an instruction replacement component operating on the different processor and in data communication with the instruction locator, the instruction replacement component adapted to receive a branch ~~replacement~~ instruction for the at least one machine instruction and operable to replace the at least one machine instruction in the executable form of the software program with the branch ~~replacement~~ instruction.

11. (original) The computer-implemented system of Claim 10 wherein the instruction locator is operable to identify an address for the specified machine instruction using the image of the executable containing the machine instructions that comprise the executable.

12. (original) The computer-implemented system of Claim 11 wherein the instruction replacement component is operable to insert the replacement instruction into a program memory image of the software program at said address.

13. cancel

14. (currently amended) The computer-implemented system of Claim 10 wherein the instruction replacement component is operable to generate a set of relocation instructions, such that the branch ~~replacement~~ instruction passes processing control to the set of relocation instructions.

15. (currently amended) The computer-implemented system of Claim 14 wherein the instruction replacement component is further operable to insert the set of relocation instructions in a memory space of the microprocessor that resides ~~outside of~~ at an unused address space of ~~for~~ the software program.

16. (new) A method for controlling the value of a RAM variable inside an executable program, comprising:

presenting a software program in executable form and having a plurality of machine instructions of a finite quantity of fixed lengths;

identifying at least two machine instructions that accesses a variable defined in random access memory associated with the software program;

replacing each of the identified machine instruction in the executable form of the software program with a branch instruction, where each branch instruction references a different address outside an address space of the software program;

defining a set of relocated instructions at each address referenced by the branch instructions, wherein each set of relocated instructions function to change a value of the variable in a different manner; and

executing the executable form of the software program having the branch instruction.

17. (new) The method of Claim 16 wherein the step of identifying at least one machine instruction further comprises determining location information for the at least one machine instruction within the software program.

18. (new) The method of Claim 17 wherein the step of determining location information further comprises identifying an address for the at least one machine instruction using the image of the executable containing the machine instructions that comprise the executable.

19. (new) The method of Claim 18 wherein the step of replacing the at least one machine instruction further comprises inserting the replacement branch instruction into a program memory image of the software program at said address.